

1 **In the Claims**

2 Claims 1, 17, 22 and 31 are amended.

3 Claims 1-34 remain in the application for consideration and are listed as
4 follows:

5
6 1. (Currently Amended) A system comprising:

7 a plurality of sources; and

8 an interface, selectively coupled to the plurality of sources, to generate and
9 implement a development project of processing chains at least one chain of which
10 comprises multiple filters, wherein the interface loads a processing chain for each
11 of the plurality of media sources at a point during the execution of the project
12 when the chain is required, and wherein the interface is configured to unload at
13 least a subset of the chains when they are not required, wherein unloading of said
14 subset is accomplished based, at least in part, on a generated execution list
15 comprising:

16 a chain identifier field which maintains a list of chains utilized in the
17 project;

18 a source identifier field which contains information denoting a
19 project source;

20 a project time field which denotes at what point during the project
21 execution of a source chain is required;

22 a source time field which denotes what portion of a source file is
23 required; and

24 a dependencies field which denotes whether an associated chain is
25 dependent on any other chains.

1
2 2. (Original) A system according to claim 1, wherein the interface is a
3 render engine, exposed to a media processing system implemented on a computing
4 system.

5
6 3. (Original) A system according to claim 1, further comprising an
7 application program, coupled to the interface, to enable a user to define a media
8 processing project.

9
10 4. (Previously Presented) A system according to claim 1, wherein the
11 interface only loads those processing chains required during the next M seconds of
12 project execution and, if the currently loaded chain-count does not exceed a
13 threshold, T, where M and T are greater than 0.

14
15 5. (Original) A system according to claim 4, wherein M is less than a
16 time required to load a processing chain.

17
18 6. (Original) A system according to claim 4, wherein if the currently
19 loaded chain-count has reached a threshold, T, the interface identifies one or more
20 currently loaded chains that can be unloaded.

21
22 7. (Previously Presented) A system according to claim 6, wherein the
23 interface identifies one or more currently loaded chains that will not be used
24 during the next N seconds to unload, where N is greater than 0.
25

1 8. (Original) A system according to claim 7, wherein M is less than N.

2
3 9. (Original) A system according to claim 7, wherein the interface
4 determines whether the identified one or more chains will be required during
5 subsequent execution of the project, or in a future project and, if so, caches the
6 identified chain(s).

7
8 10. (Previously Presented) A system according to claim 7, wherein the
9 interface determines whether unloading of the identified one or more chains
10 reduces the chain-count below a maximum allowable threshold, V, and, if so,
11 loads the chains required in the next M seconds, wherein V is greater than 0.

12
13 11. (Previously Presented) A system according to claim 9, whereupon
14 determining that the chain-count is not below V, the interface identifies one or
15 more lowest priority chains and unloads the identified chain(s), wherein V is
16 greater than 0.

17
18 12. (Original) A system according to claim 10, wherein the interface
19 removes the identified chains from the active project and caches the removed
20 chains.

21
22 13. (Original) A system according to claim 10, wherein the interface
23 loads the chains required during the next M seconds.

24
25 14. (Original) A system according to claim 6, wherein the interface

1 unloads a chain when all matrix switch filter(s) of the filter graph provide an
2 indication that the chain is no longer required.

3
4 15. (Original) A system according to claim 4, wherein T is set to one (1).

5
6 16. (Original) A system according to claim 15, whereby setting T equal
7 to one (1), the interface will be required to search for and potentially unload chains
8 which are not required to support execution of the project for the next N seconds.

9
10 17. (Currently Amended) A computer-implemented method for
11 generating and managing a development project, the method comprising:

12 identifying processing chains required to support execution of the
13 development project over the next M seconds; and

14 loading the identified processing chains as long as a currently loaded chain-
15 count does not exceed an initial threshold, T, wherein T and M are greater than 0;

16 and

17 unloading a subset of loaded processing chains based, at least in part, on a
18 generated execution list comprising:

19 a chain identifier field which maintains a list of chains utilized in the
20 project;

21 a source identifier field which contains information denoting a
22 project source;

23 a project time field which denotes at what point during the project
24 execution of a source chain is required;

25 a source time field which denotes what portion of a source file is

1 required; and

2 a dependencies field which denotes whether an associated chain is
3 dependent on any other chains.

4
5 18. (Previously Presented) A method according to claim 17, further
6 comprising:

7 identifying currently loaded chains that will not be used during the next N
8 seconds, wherein N is greater than 0; and

9 removing the identified chains from the development project.

10
11 19. (Previously Presented) A method according to claim 18, further
12 comprising:

13 determining whether the chain-count has dropped below a maximum
14 allowable chain-count, V, after removing the identified chains from the
15 development project, wherein V is greater than 0;

16 identifying one or more low priority chains and removing them from the
17 development project; and

18 loading the chains required in the next M seconds.

19
20 20. (Original) A method according to claim 18, wherein removing the
21 identified chains comprises:

22 determining whether the identified chains will be required during
23 subsequent execution of the development project or future development projects;
24 and

25 caching the identified chains if they will be used during subsequent

1 execution of the development project and/or future projects.

2
3 21. (Original) A method according to claim 17, wherein T is set to one
4 (1) such that an implementing media processing system always attempts to unload
5 unused chains prior to loading chains.

6
7 22. (Currently Amended) A computer-implemented method for
8 managing a media processing project, the method comprising:

9 identifying each of a plurality of sources required to satisfy the media
10 processing project;

11 determining when one or more chain(s) associated with each of the
12 plurality of sources is required to support execution of the media processing
13 project; and

14 selectively loading and unloading each of the chains during execution of
15 the filter graph based, at least in part, on when each of the chains are required to
16 support execution of the media processing project, at least some selectively loaded
17 and unloaded chains comprising multiple filters, wherein unloading of said chains
18 is accomplished based, at least in part, on a generated execution list comprising:

19 a chain identifier field which maintains a list of chains utilized in the
20 project;

21 a source identifier field which contains information denoting a
22 project source;

23 a project time field which denotes at what point during the project
24 execution of a source chain is required;

25 a source time field which denotes what portion of a source file is

1 required; and

2 a dependencies field which denotes whether an associated chain is
3 dependent on any other chains.

4
5 23. (Previously Presented) A method according to claim 22, wherein
6 loading and unloading chains comprises:

7 identifying which processing chain(s) will be required within the next M
8 seconds of project execution; and

9 loading the identified processing chain(s) if a currently loaded chain-count
10 does not exceed a threshold, T, wherein M and T are greater than 0.

11
12 24. (Previously Presented) A method according to claim 23, further
13 comprising:

14 identifying one or more processing chains that will not be required in the
15 next N seconds if the chain-count threshold T has been reached, wherein N is
16 greater than 0; and

17 removing the identified one or more processing chains from the processing
18 project.

19
20 25. (Original) A method according to claim 24, wherein removing the
21 identified one or ore processing chains comprises:

22 determining whether the identified processing chains will be required
23 during subsequent execution of the media processing project, or a future
24 processing project; and

25 caching at least a subset of the processing chains if they will be required

1 during subsequent execution of the media processing project, or future processing
2 project(s).

3
4 26. (Previously Presented) A method according to claim 24, further
5 comprising:

6 determining whether removing one or more of the identified processing
7 chains reduces the chain-count below a maximum allowable threshold, V, wherein
8 V is greater than 0; and

9 identifying one or more low priority processing chains to remove to reduce
10 the chain-count below the maximum allowable threshold if it is determined that
11 the chain-count exceeds the maximum allowable threshold V.

12
13 27. (Original) A method according to claim 26, wherein V is greater
14 than T.

15
16 28. (Original) A method according to claim 23, wherein T is set to one
17 (1).

18
19 29. (Original) A storage medium comprising a plurality of executable
20 instructions which, when executed, implement a method according to claim 22.

21
22 30. (Original) A computing system comprising:
23 a storage medium having stored therein a plurality of executable
24 instructions; and
25 an execution unit, coupled to the storage medium, to execute at least a

1 subset of the plurality of executable instructions to implement a method according
2 to claim 22.

3
4 31. (Currently Amended) A storage medium comprising a plurality of
5 executable instructions which, when executed, implements an interface to manage
6 development and execution of a development project, wherein the interface
7 identifies processing chains required to support execution of the development
8 project over the next M seconds, and loads the identified processing chains as long
9 as a currently loaded chain-count does not exceed an initial threshold, T, wherein
10 M and T are greater than 0, and wherein the interface unloads chains based, at
11 least in part, on a generated execution list comprising:

12 a chain identifier field which maintains a list of chains utilized in the
13 project;

14 a source identifier field which contains information denoting a
15 project source;

16 a project time field which denotes at what point during the project
17 execution of a source chain is required;

18 a source time field which denotes what portion of a source file is
19 required; and

20 a dependencies field which denotes whether an associated chain is
21 dependent on any other chains.

22
23 32. (Previously Presented) A storage medium according to claim 31,
24 wherein the interface identifies one or more processing chains that will not be
25 required to support execution of the development project over the next N seconds,

1 and removes such chains from the development project, wherein N is greater than
2 0.

3
4 33. (Original) A storage medium according to claim 32, wherein T is set
5 equal to one (1) to force the interface to identify chains to unload before loading
6 each chain required to support processing of the development project over the next
7 M seconds.

8
9 34. (Original) A storage medium according to claim 32, wherein the
10 interface determines whether removed processing chains will be required in
11 support of subsequent processing of the development project and/or a future
12 development project and, if so, caches the removed processing chains for
13 subsequent use.